

week 5 design exercise

Tasks

1. Write 250 words that critically reflect on this week's reading:
McLuhan, M. and Q. Fiore. 1967. The Medium is the Massage: An Inventory of Effects. London: Corte Madera, Gingko Press.
2. Create an arduino-based system that translates temperature values into light.

Key Terms

pulse width modulation (PWM)

A type of digital signal that allows for more sophisticated control of electronic circuits. Typically a digital signal can only be on (HIGH, 1) or off (LOW, 0). However, pulse width modulation (PWM) allows for gradations between on and off by flickering a pin signal on and off at high frequency to create the effect of a dimming LED or to control a motor's spin rate or position. With PWM, you can create complex effects.

map(value, ALow, AHigh, ZLow, ZHigh)

Map is a function that 're-maps', or proportionally changes, one set of values to another. In other words a value of ALow would output as ZLow. A value of AHigh would output as ZHigh. All the values in-between would be proportionately output from ZLow to ZHigh.

Ex: To proportionally map a variable with a value 5 to output proportionally between 100 and 1000, you would require the following code:

```
int someVariable = 5;           // the variable 'someVariable' has a value of 5
int mappedVariable;           // create new variable called 'mappedVariable'

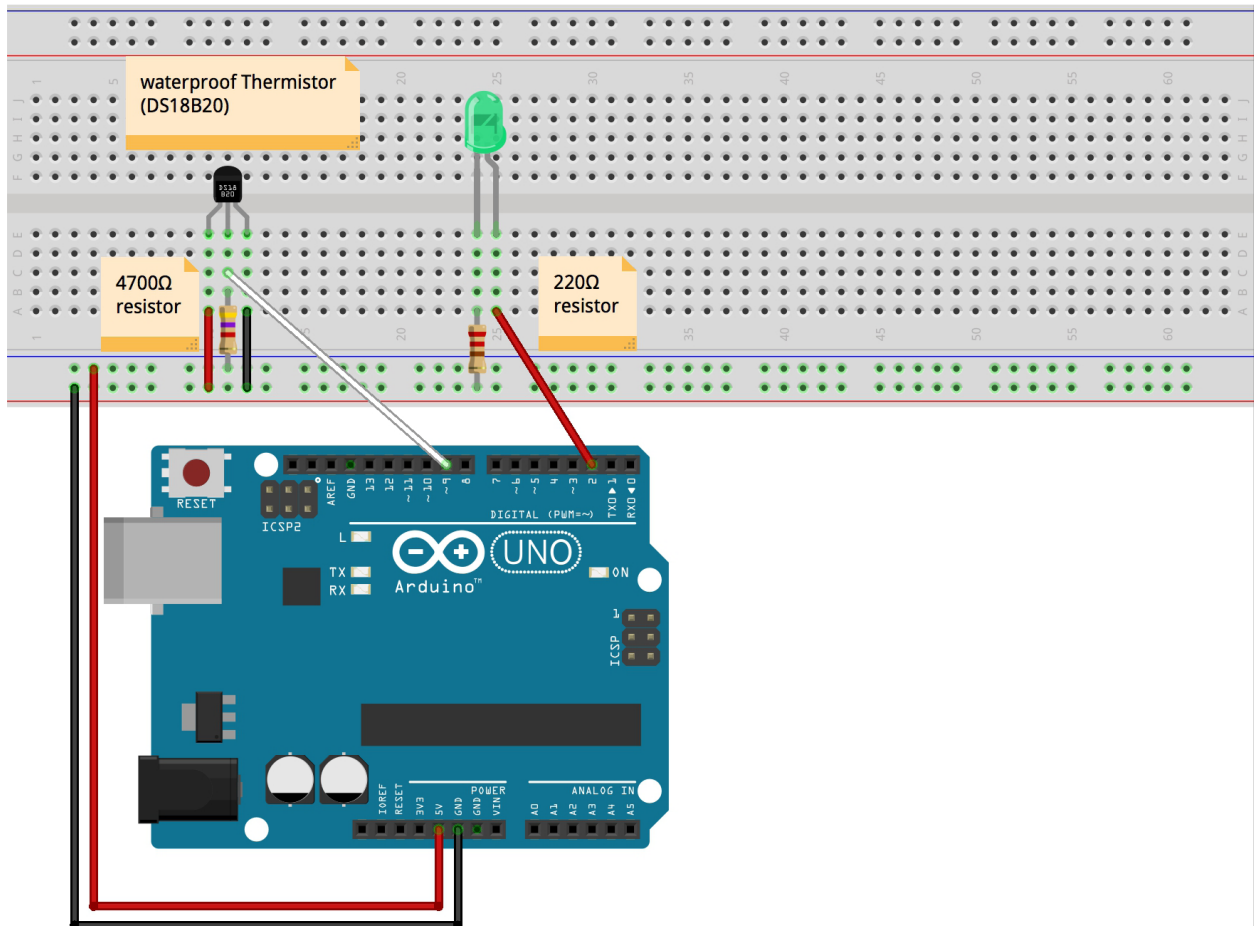
// store a new mapped value in variable 'mappedVariable' that proportionally
// 'maps' the value of variable named 'someVariable' in a set from 1 and to 10
// to a set between 100 and 1000.
mappedVariable = map(someVariable, 1, 10, 100, 1000);
```

The output of mappedVariable is 500.

Wiring Setup – Thermistor

A thermistor is a sensor to determine local temperature. The sensor acts like a resistor and changes the input voltage based on the temperature it receives.

First, connect the thermistor (part no: ds18b20) to the Arduino using the diagram below. Then connect your LED to the Arduino. Make sure to connect the appropriate resistors in this circuit.



Arduino – Installing Libraries

Then, navigate to: LMS > IDES3010 > Devon Ward > Week 5 and download the two libraries which are required to run the temperature sensor. The libraries are called: `OneWire` and `DallasTemperature`. In the Arduino IDE, install both of these libraries by selecting from the top toolbar: Sketch > Include Library > Add .ZIP Library ... and select the downloaded .zip files.

Programming - Altogether This Time

Then, write the following code which will translate the temperature reading from the temperature sensor to a value that will change the brightness of your LED using PWM.

```
#include <OneWire.h>           // library required for the temperature sensor (DS-18B20)
#include <DallasTemperature.h> // library required to read temperature from the sensor

// Data wire from temp. sensor is plugged into pin 9 on the Arduino
#define ONE_WIRE_BUS 9
// Setup a oneWire instance to communicate with your temp. sensor
OneWire oneWire(ONE_WIRE_BUS);
// Pass our oneWire reference value to Dallas Temperature to convert to celsius
DallasTemperature sensors(&oneWire);

int pwmPin = 2;                // variable for you LED pin
int temp = 0;                  // variable to store the temperature value
int light = 0;                 // variable to store the light brightness value

void setup(void)
{
    Serial.begin(9600);        // Begin serial communication
    sensors.begin();           // initialises your temp. sensor
    pinMode(pwmPin, OUTPUT);    // sets your pwmPin as an output pin
}

void loop(void)
{
    sensors.requestTemperatures(); // sends the command to get temperatures
    temp = sensors.getTempCByIndex(0); // store the sensor values in variable 'temp' as celsius
    light = map(temp, 10, 40, 0, 255); // maps 'temp' value from 10C to 40C as digital value from 0 to 255
    analogWrite(pwmPin, light);        // analogWrite is a function for PWM that outputs 'light' value to 'pwmPin'
    Serial.print("Temperature is: "); // print 'Temperature is: ' to Serial Monitor
    Serial.println(temp);              // print 'temp' value to Serial Monitor
    Serial.print("PWM: ");             // print 'PWM: ' to Serial Monitor
    Serial.println(light);             // print 'light' value to Serial Monitor
    delay(500);                       // delay for 500ms
}
```

Upload your sketch to your Arduino. Open the Serial Monitor to view the temperature values and the light brightness values. Test that the sensor translates temperature to light by holding the temperature sensor in your hands until the value increases. You should be able to see the brightness increase as well.

Record your video

Record a 30-second video that shows:

1. your hardware connection
2. your final sketch
3. the Serial Monitor output of your final sketch
4. the functioning system

Create a .zip file that contains your 30-second video and your written text about this week's reading. Upload your files to the dropbox link which will be provided at the start of next class. You cannot receive credit for this assignment if you are not present in class.
